

# ADAPTIVE DISTRIBUTED ALGORITHMS FOR POWER-EFFICIENT DATA GATHERING IN SENSOR NETWORKS

*Jugoslava Aćimović<sup>1</sup>, Baltasar Beferull-Lozano<sup>2</sup> and Răzvan Cristescu<sup>3</sup>*

<sup>1</sup>Laboratory of Nonlinear Systems (LANOS), EPFL, Lausanne CH-1015, Switzerland

<sup>2</sup>Audiovisual Communications Laboratory (LCAV), EPFL, Lausanne CH-1015, Switzerland

<sup>3</sup>Center for Mathematics of Information, Caltech, 136-93 Pasadena, CA 91125, USA

## ABSTRACT

In this work, we consider the problem of designing adaptive distributed processing algorithms in large sensor networks that are efficient in terms of minimizing the total power spent for gathering the spatially correlated data from the sensor nodes to a sink node. We take into account both the power spent for purposes of communication as well as the power spent for local computation. Our distributed algorithms are also matched to the nature of the correlated field, namely, for piecewise smooth signals, we provide two distributed multiresolution wavelet-based algorithms, while for correlated Gaussian fields, we use distributed prediction based processing. In both cases, we provide distributed algorithms that perform network division into groups of different sizes. The distribution of the group sizes within the network is the result of an optimal trade-off between the local communication inside each group needed to perform decorrelation, the communication needed to bring the processed data (coefficients) to the sink and the local computation cost, which grows as the network becomes larger. Our experimental results show clearly that important gains in power consumption can be obtained with respect to the case of not performing any distributed decorrelating processing.

## 1. INTRODUCTION

Consider a network of sensors that measure certain data and that have to transmit all obtained information to a common central node, or sink, for processing or storage. Usually, in sensor networks, since the sensors are strongly constrained in terms of battery power [4], a meaningful task in such settings is to design algorithms for data transmission that minimize the total cost of gathering the measured data [1]. The simplest possible strategy consists of direct transmission of the information from all sensors to the sink (raw

data gathering). However, it is clear that even for moderately sized networks this approach results in very inefficient use of the power for two main reasons. First, the measured phenomenon is typically a spatially correlated process, and consequently, physically close nodes measure correlated spatial samples, and this similarity can be exploited to decrease the amount of information required to be transmitted. Second, further improvements can be obtained, in the context of wireless sensor networks, by using multi-hop communication, having sensors that relay their data via neighboring nodes rather than sending the data directly to the sink.

In this work, we focus on two classes of fields: (a) deterministic signals for which wavelet processing is appropriate, namely piecewise defined signals, such as for instance piecewise constant signals, which can be often found in practice after some threshold detection process, and (b) correlated Gaussian fields, for which prediction based processing is the optimal way to perform decorrelation across the nodes. For deterministic piecewise continuous signals, the amount of correlation present in the signal is essentially determined by the number of discontinuities. We analyze how the number of discontinuities affects the total transmission cost when signal decorrelation is employed for data gathering, in comparison to raw data gathering. A joint consideration of wavelet coding and power efficient transmission is exploited in [2], where wavelet lifting is used. However, there are two important differences of our work with respect to [2]: a) the signal models we consider, which motivate the distributed algorithms that we provide, are different to the ones considered in [2], b) the network model considered in [2] is different, more specifically, in [2], the sink is not located at the extremity of the network.

We analyze several efficient algorithms for data gathering based on distributed signal processing, adapted to the nature of the measured data. For the class of piecewise constant signals, we propose an adaptive algorithm that involves an adaptive network segmentation into groups of variable size, where the segmentation is adapted to the particular signal or realization of the field. For correlated Gaussian fields, we propose an algorithm based on performing pre-

---

THIS WORK WAS SUPPORTED (IN PART) BY THE NATIONAL COMPETENCE CENTER IN RESEARCH ON MOBILE INFORMATION AND COMMUNICATIONS SYSTEMS (NCCR-MICS), SUPPORTED BY THE SWISS NATIONAL SCIENCE FOUNDATION UNDER GRANT NUMBER 5005-67322.

diction locally inside each group, finding also the optimal group sizes. In both cases, first, some local processing takes place inside each group, and second, the processed data is sent to the sink node in a multi-hop fashion. Multi-hop transmission (as opposed to direct transmission from sensor to sink) is clearly motivated by the fact that in sensor networks with wireless communication, the power dissipated in transmitting  $R$  bits through a distance  $d$  is proportional to  $R \cdot d^\alpha$ , where  $\alpha > 1$  [5]. We show by numerical simulations that our algorithms provide important power gains for the data gathering. Moreover, we show how these power gains change, as a function of the number of discontinuities present in the signal, for piecewise constant signals, and as a function of the amount of correlation between samples in the case of random correlated fields.

The rest of the paper is structured as follows. In Section 2, we describe our sensor network scenario and assumed signal models. In Section 3, we introduce the algorithms for wavelet based data gathering using the Haar wavelet transform, where we describe our adaptive distributed algorithm. Section 4 presents our adaptive algorithm for data gathering of random correlated Gaussian fields. Finally, in Section 5, we present several simulation results.

## 2. PROBLEM STATEMENT

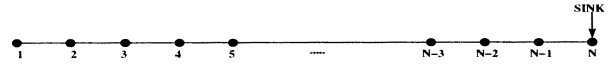
### 2.1. Network Model

In this work, we consider, for the sake of simplicity, a one-dimensional network model<sup>1</sup>. The one-dimensional models simplifies the routing problem while still keeping the essential characteristics of the two dimensional networks, in terms of analyzing certain behaviors of large sensor networks. We consider a network of  $N = 2^M$  sensors ( $M$  being an integer) placed on a line (see Fig. 1). The network sink (we assume that the node  $N$  in Fig. 1 collects all measured data) is on the extreme right end of the network and we denote by  $d$  the distance between each pair of neighboring sensors. We assume a multi-hop data transmission model, motivated by the power efficiency constraints on the network nodes and the usual power cost for wireless transmission [5], namely, the transmission cost from one sensor to its neighbor is given by  $[\text{rate}] \times [\text{path weight}]$ , where  $[\text{rate}]$  is the number of transmitted bits, and  $[\text{path weight}]$  is a power (usually 2 or 4) of the inter-node distance  $d$ . Due to the presence of correlation in the sensed data, there are two types of communication required: local communication to remove correlation in the data and communication from sensors to sink to bring the processed data to the sink.

### 2.2. Model of Measured Signal and Data Processing

In this work, we propose several efficient data gathering algorithms for two classes of fields, namely, deterministic

<sup>1</sup>In our current work, we are extending the results of this paper to two-dimensional network models.



**Fig. 1.** A one-dimensional network model. The sink, represented by the node  $N$ , is placed at the right extremity of the network

continuous piecewise polynomial signals (in particular, we focus on constant piecewise signals), and also spatially distributed Gaussian correlated processes. In either case, the whole set of spatial sample measurements is represented by a vector  $\mathbf{Y} = (Y_1, Y_2, \dots, Y_N)$  where  $Y_i$  corresponds to the sample obtained by sensor  $i$ . Our goal is to transport the samples  $\{Y_i\}_{i=1}^N$  from the sensors to the sink in such a way as to minimize the total power cost spent. We assume that the quantization (necessary for the subsequent digital transmission) is sufficiently fine so that the resulting quantization distortion is very small and the error incurred in performing decorrelating operations with quantized samples is small.

#### 2.2.1. Continuous Piecewise-Constant Fields

The first class of signals we consider are deterministic piecewise constant signals with a finite number of discontinuities. Between any two discontinuities, the signal takes a value from the finite interval  $[a, b]$ , with  $a, b$  real numbers. We assume that the positions of the discontinuities follow a uniform distribution over the interval  $[1, N]$ , i.e. the interval of all possible locations in the network, but we also impose a constraint that ensures having a sufficiently small frequency of discontinuities, so that between any two neighbor sensors, there can be at most one discontinuity. We propose a distributed adaptive algorithm (described in Section 3.2.1) based on the Haar transform, which is known to provide a good data representation in the case of piecewise constant signals. The standard full-size Haar transform consists of  $M$  levels for a set of  $N = 2^M$  samples. Denote the low-pass (LP) and high-pass (HP) coefficients as  $Y^{k+1} = (Y_1^{k+1}, Y_2^{k+1}, \dots, Y_{\frac{N}{2^k}}^{k+1})$  and  $D^{k+1} = (D_1^{k+1}, D_2^{k+1}, \dots, D_{\frac{N}{2^k}}^{k+1})$ , respectively. The computation is done in the usual way:

$$Y_i^{k+1} = \frac{Y_{2i-1}^k + Y_{2i}^k}{\sqrt{2}}, \quad D_i^{k+1} = \frac{Y_{2i-1}^k - Y_{2i}^k}{\sqrt{2}} \quad (1)$$

for  $k = 1, 2, \dots, M$  and  $i = 1, 2, \dots, \frac{N}{2^k}$ . Notice that if the process has a single discontinuity, then there is at most one non-zero HP coefficient in each multiresolution level  $k$ , and this results in at most  $M$  non-zero HP coefficients for the whole representation, with the rest of coefficients being zero. Generally, for a signal with  $P$  discontinuities, an upper bound for the number of non-zero HP coefficients is given by  $\min(P \cdot M, N)$ . If the number of discontinuities is small enough, the set of HP coefficients will contain a large number of zeros, and by spending just one bit (e.g. first bit transmitted) to sign the presence of a zero coefficient, we

can achieve an important amount of power saving. However, coding a non-zero coefficient requires a much bigger bit rate. Thus, we represent each coefficient with a packet of bits where an initial bit indicates the coefficient presence and the remaining bits contain the information about its value. In this work, we show that when the number of discontinuities is small enough, the creation of zero-valued coefficients provides an important saving in the total power spent. The same quantization stepsize  $\Delta$  is used for quantizing the data to be transmitted and no entropy coding is applied for this class of deterministic signals. Therefore, it is clear that all the LP coefficients at level  $k$  will be coded with the same rate, which we denote by  $R_Y(k)$ . That is, since  $a \cdot 2^{\frac{k-1}{2}} \leq Y_i^k \leq b \cdot 2^{\frac{k-1}{2}}$ ,  $i = 1, 2, \dots, \frac{N}{2^k}$ , we have that  $R_Y(k) = \log_2 \left( \frac{b-a}{\Delta} 2^{\frac{k-1}{2}} \right) + 1 = R_Y(1) + \frac{k-1}{2}$ , where  $R_Y(1) = c + 1$  and  $c = \left\lceil \log_2 \left( \frac{b-a}{\Delta} \right) \right\rceil$  is the rate necessary to code the original samples at the 0-th level. Similarly, it is straightforward to see that for all the non-zero HP coefficients, the necessary rate at each level  $k$ , denoted by  $R_D(k)$ , is also given by  $R_D(k) = R_Y(1) + \frac{k-1}{2}$ .

### 2.2.2. Correlated Random Gaussian Fields

Given the sample vector  $\mathbf{Y}$ , the correlation model is determined by the covariance matrix  $\mathbf{K}$ . Assuming a zero-mean process, the  $N$ -dimensional multivariate normal distribution is given by  $f(\mathbf{Y}) = \frac{1}{\sqrt{2\pi \cdot \det(\mathbf{K})}^{1/2}} \cdot e^{-\frac{1}{2} \mathbf{Y}^T \mathbf{K}^{-1} \mathbf{Y}}$ . In this work, we use the following correlation models:

$$\mathbf{K}_{i,j} = \sigma^2 \cdot e^{-\alpha(|i-j| \cdot d)^\beta} \quad (2)$$

where  $\beta \in \{1, 2\}$ , and  $\alpha$  determines the speed at which the correlation decreases. For these signals, the best decorrelating technique that can be used is to perform prediction based on neighbor samples (similar to DPCM). In other words, we code each signal  $Y_i$  using the knowledge about a certain number  $n$  of neighbor signals on the line, namely, using  $Y_{i+1}, Y_{i-2}, \dots, Y_{i-n}$ . However, notice that since the sampled signals need to be quantized prior to their transmission, the prediction, as usual, will be based on reconstructed signals  $\hat{Y}_{i-1}, \hat{Y}_{i-2}, \dots, \hat{Y}_{i-n}$ . The prediction is given by:

$$\hat{Y}_i = \mathbf{K}_{Y_i; \hat{Y}_{i-1}, \dots, \hat{Y}_{i-n}} \cdot \mathbf{K}_{\hat{Y}_{i-1}, \dots, \hat{Y}_{i-n}}^{-1} \cdot [\hat{Y}_{i-1}, \dots, \hat{Y}_{i-n}]^T \quad (3)$$

and the prediction error is given by  $\epsilon_i = Y_i - \hat{Y}_i$ , which has substantially smaller variance and thus requires less bits to be coded; notice that this residual has to be also quantized for transmission, obtaining  $\hat{\epsilon}_i$ . The network sink will obtain the reconstruction of the data  $\hat{Y}_i$  measured by the  $i$ -th node simply as  $\hat{Y}_i = \hat{Y}_i + \hat{\epsilon}_i$ . Notice that when assuming high-rate quantization, i.e. a small quantization stepsize  $\Delta$ , followed by entropy coding at the relay nodes, to code a sample with variance  $\sigma^2$ , the required bit rate is well approximated by

$\log_2 \left( \frac{\sqrt{2 \cdot \pi \cdot e \cdot \sigma^2}}{\Delta} \right)$  bits. However, this important reduction in variance comes together with a substantial increase in computational complexity, since as the prediction is performed by conditioning on more and more nodes, the number of local computations in (3) that have to be performed increase also rapidly. In Sections 4 and 5, we show that when the network is large enough, there exists an optimal network segmentation, which minimizes the total power cost, into groups of sensors where the decorrelation is only performed throughout each group. On the other hand, in the context of piece-wise constant signals, the computational complexity spent in a Haar wavelet step is very small since each LP and HP coefficient involve just 1 addition and 1 multiplication.

### 2.3. Power Cost of Data Gathering

We define the cost function for our data gathering problem as the total power consumption taking place in the network for collecting data to the sink. In our total power cost, in addition to considering communication cost, in terms of both local communication to perform decorrelation and communication from sensors to sink, we also introduce the power spent for local computation, in terms of power dissipated in the DSP circuits. For processor-dependent power consumption models, we use values described in the literature [5].

In the usual literature of sensor networks, this cost is usually ignored, as compared to the cost of communication. However, we show in this work that if the sensor network becomes dense enough, in some cases, namely for correlated Gaussian fields, because of the increasing need for performing decorrelation by means of computing predictions, the cost of local computation may influence importantly the design of the most efficient distributed algorithm for performing the data gathering. For piecewise constant signals, the impact of computation is very small due to the simplicity of Haar transform computing. Denote the bit-rate for coding the data  $\xi$  as  $R(\xi)$  and the number of hops between a transmitter  $X$  and receiver  $Z$  as  $k_{X \rightarrow Z}$ . Also, denote the dissipation in the DSP circuits as  $\text{cost}_{DSP}$ . Then, the cost function for sending the data between  $X$  and  $Z$  is:

$$\begin{aligned} \text{cost}(X \rightarrow^\xi Z) &= \text{cost}_{COM} + \text{cost}_{DSP} \\ &= R(\xi) \cdot k_{X \rightarrow Z} \cdot d^2 + \mu \cdot \text{num}_{OP} \\ &= (R(\xi) \cdot k_{X \rightarrow Z} + \gamma \cdot \text{num}_{OP}) \cdot d^2 \end{aligned} \quad (4)$$

where the parameter  $\gamma = \frac{\mu}{d^2}$  can be interpreted as the ratio between the power dissipated by the processor (at the transmitter) in performing a single operation and the power necessary for transmitting one bit over the unit distance, including the power dissipated by both the transmitter and the receiver; this value will depend on the particular processor that is used [5]. Throughout this work, we compare the performance of our algorithms with the performance of a reference raw data gathering algorithm in which each sensor simply sends (by using multi-hops) each measured signal to the

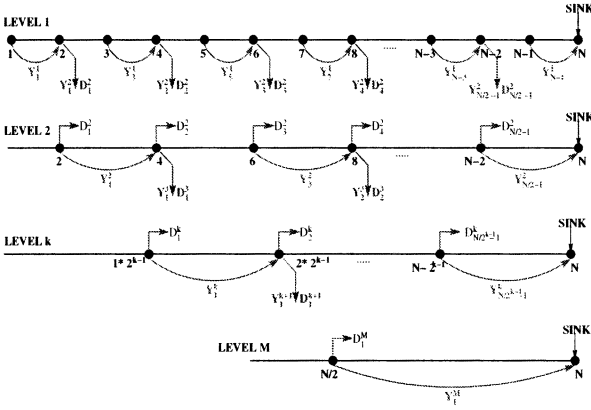


Fig. 2. Full-size non-adaptive Haar multiresolution processing.

sink without any local decorrelating processing. For this reference algorithm, the total gathering cost for a network with  $N$  sensors is  $\text{cost}_{ref}(N) = c \frac{N \cdot (N-1)}{2}$  ( $\text{cost}_{DSP} = 0$ , since we do not apply any signal processing strategy for decorrelation) and all the transmitted data are non-zero. Thus  $\text{cost}_{ref}(N) = O(N^2)$ . For the evaluation of the performance of an algorithm, we compute the relative cost as:

$$\eta(N) = \frac{\text{cost}(N)}{\text{cost}_{ref}(N)}. \quad (5)$$

where  $\text{cost}(N)$  is given by the sum of (4) over all transmitter-receiver pairs over all the steps of the data gathering algorithm. Notice that this relative cost (5) is independent of the inter-node distance because of the quotient. Therefore, from now on, throughout this paper and without loss of generality, we consider  $d = 1$ . Notice that the desired behavior of an algorithm is to have  $\eta(N) < 1$  and as small as possible.

### 3. DATA GATHERING ALGORITHMS FOR PIECEWISE CONSTANT SIGNALS

#### 3.1. Distributed Full Haar Algorithm

First, we describe a basic distributed algorithm based on the full Haar transform that we call *NetHaar*, which provides insight into the design of the two adaptive distributed algorithms that we present. For a network of  $N = 2^M$  nodes, *NetHaar* consists of  $(M + 1)$  steps, as illustrated in Fig. 2.

Let  $Y^1$  denote the original measured data. In the first algorithm step, sensors with the odd indices send their data to the sensors with even indices. Thus, the set of transmitted signals is  $(Y_1^1, Y_3^1, \dots, Y_{N-1}^1)$ . Then, sensors 2, 4, ...,  $N$  calculate the LP and HP coefficients by combining the received data and the data measured by themselves, obtaining  $Y^2 = (Y_1^2, Y_2^2, \dots, Y_{N/2}^2)$  and  $D^2 = (D_1^2, D_2^2, \dots, D_{N/2}^2)$ , respectively. We can ignore the odd index sensors in the next steps of the algorithm, since the remaining sensors contain all information necessary for recovering the original data. For the  $k$ -th step of the algorithm, nodes  $2^{k-1} \cdot 1$ ,

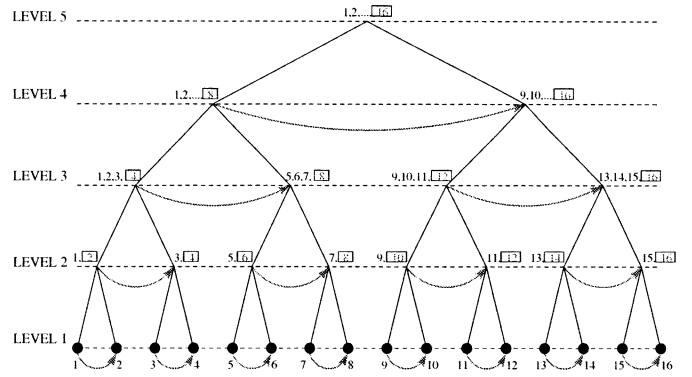


Fig. 3. Tree representation of the algorithms.

$2^{k-1} \cdot 3, \dots, N - 2^{k-1}$  send their coefficients to the nodes  $2^{k-1} \cdot 2, 2^{k-1} \cdot 4, \dots, N$ , and the new set of coefficients is calculated as in (1), for  $k = 1, 2, \dots, M$  and  $i = 1, 2, \dots, \frac{N}{2^k}$ . Finally, in the  $M + 1$ -th step of the algorithm, node  $N$  sends the last calculated coefficients,  $Y_1^{M+1}$  and  $D_1^{M+1}$ , to the network sink. Two types of transmissions occur, namely, sensor-to-sensor and sensor-to-sink transmission. Also, additional cost is introduced as a result of signal processing in the network nodes. Thus, the total cost is:

$$\text{cost}(N) = (\text{cost}_{LP}(N) + \text{cost}_{HP}(N)) + \gamma \cdot \text{num}_{OP}(N). \quad (6)$$

- total sensor-to-sensor communication  $\text{cost}_s(N)$ :

$$\text{cost}_s(N) = \sum_{k=1}^M \sum_{i=1}^{\frac{N}{2^k}} R(Y_i^k) 2^{k-1} d^2;$$

- total sensor-to-sink communication cost  $\text{cost}_{sd}$ :

$$\text{cost}_{sd}(N) = \sum_{k=2}^{M+1} \sum_{i=1}^{\frac{N}{2^{k-1}}} R(D_i^k) (N - 2^{k-1}i + 1) d^2 + R(Y_1^{M+1}) d^2$$

since in the  $k^{\text{th}}$  step, each transmitting sensor sends its LP coefficient over  $2^{k-1}$  hops and its HP coefficient over  $N - 2^{k-1}i + 1$  hops.

- total signal processing  $\text{cost}_{DSP}$ , which is equal to the number of performed operations<sup>2</sup>:

$$\text{num}_{OP}(N) = 2 \cdot (N - 1).$$

The influence of the local processing cost term is given by the parameter  $\gamma$ . It should be noted that for the algorithms involving Haar processing, the total computational cost  $\text{cost}_{DSP}$  depends linearly on the number of network nodes, the cost of signal processing becomes really negligible as compared to the cost of data transmission.

<sup>2</sup>By operation we assume one addition and one multiplication, i.e. the mathematical operation necessary for calculating one Haar coefficient.

### 3.2. Data Gathering with Network Segmentation

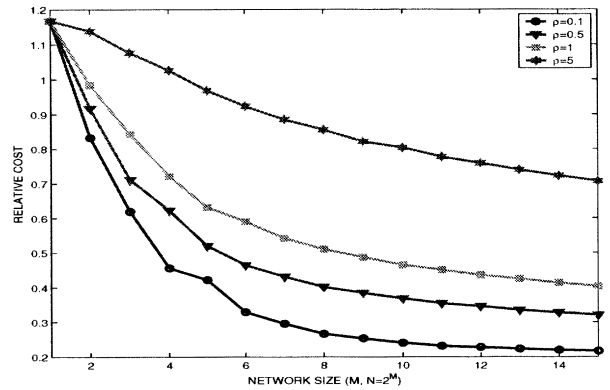
In this section, we introduce an adaptive algorithm that makes use of the data gathering strategy described in Section 3.1, but optimizing the trade-off between local communication among sensors for decorrelation, the communication from sensors to the sink, as well as the amount of local processing. As a result, the network needs to be segmented into groups, so that initially, the local communication takes place only among the sensors in each (independent) group. For the case of Haar processing, the dominant effect that determines the group size is the distribution of discontinuities in the measured process (i.e. the number of non-zero HP coefficients, as well as their position inside the network), since the involved computational complexity for each step is small. A subset of sensors is called a *group* if one of the sensors, the *leader-node*, located at the right extremity of the group, collects all necessary local information to represent the signal data measured by the sensors in the spatial region covered by the group. We illustrate all possible network segmentations as well as the communication between the corresponding groups, by using a full binary tree (see Fig. 3). Each tree leaf corresponds to a single sensor, while the root represents the whole network. Each in-tree node represents a group of sensors. Notice that the possible sizes for the groups are of the form  $2^k$ , where  $k \in \{0, 1, \dots, M\}$ , which is necessary to be able to apply the  $(k+1)$ -level Haar transform step inside each group.

Inter-group communication is allowed only between neighboring groups, as shown in Fig. 3. The *NetHaar* algorithm fully spans the tree from bottom to top, without checking for optimality of this operation in terms of cost. By using the adaptive algorithm proposed in this section, we obtain an improvement over *NetHaar* because we allow to stop adaptively at different levels of the tree, optimizing (6).

#### 3.2.1. Adaptive Algorithm: Optimal Data Gathering

In this section, we propose a fully decentralized algorithm, called *AdaptNetHaar*, which provides the optimal network division adapting its behavior to each measured signal.

The algorithm starts from the bottom of the binary tree in Fig. 3. As in *NetHaar*, the odd index sensors send their data to the even index sensors (represented by arrows in Fig. 3). Each receiving sensor makes a decision about whether to create a 2-sensor group or to keep the present 1-sensor group state, by comparing the costs of data gathering for the new group with the sum of costs corresponding to the existing 1-sensor groups. After this operation, sensors 2, 4,  $\dots$ ,  $N$  have information about their neighbors with odd index. In the second step, they can transmit data further to the leader-nodes of 4-sensor groups (see Fig. 3). A new set of coefficients is computed at the leader-nodes (precisely, two new coefficients) and the newly computed cost is compared to the cost of the previous step. The decision about collapsing the groups into one larger group is essentially deter-



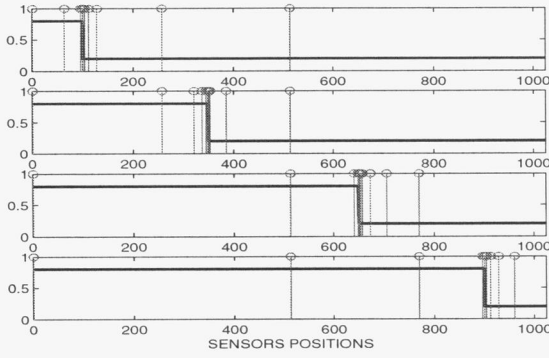
**Fig. 4.** Relative cost for *AdaptNetHaar* as a function of network size and number of discontinuities (parameter  $\rho$ ).

mined by the values of the computed HP coefficients. The algorithm proceeds similarly for the other tree levels. For  $k = 2, \dots, M+1$ , there are  $\frac{N}{2^{k-1}}$  groups of size  $2^{k-1}$ . For each group, the total cost of data gathering is compared with the gathering cost for that group corresponding to the previous algorithm steps. Further grouping is accepted only if it decreases the cost. The algorithm stops once all data reach the network sink. Note that the inter-group communication in this algorithm does not incur any additional cost with respect to *NetHaar*.

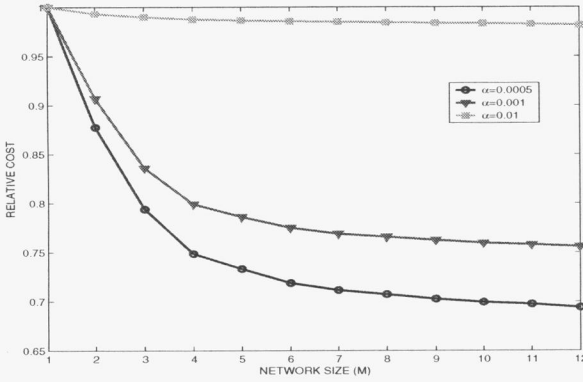
## 4. DATA GATHERING OF CORRELATED RANDOM FIELDS

For random Gaussian correlated fields, we use prediction based processing instead of Haar processing. Inside each group, the prediction processing will be done iteratively from left to right, thus, the right-hand side leader-node in each group will get all the coefficient information of that corresponding group. Notice that in this case, as opposed to piece-wise constant fields, the required processing has a computational complexity that rapidly increases with the number of samples used for computation of the residuals. Therefore, in this case, it becomes an important factor that influences the optimal sizes of the groups across the network.

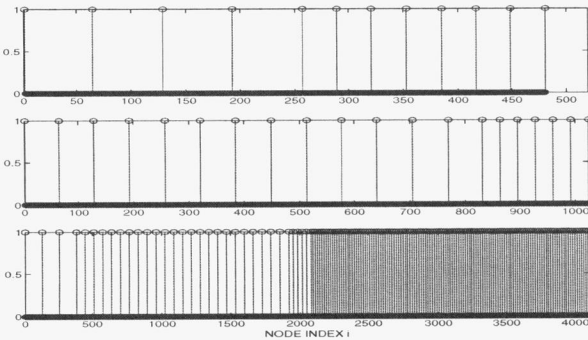
For a given correlation model, it is also possible to design an adaptive algorithm that will provide a network segmentation trading-off the communication and local processing costs. Consider a group of  $n+1$  nodes,  $i, i+1, \dots, i+n$  with node  $i+n$  being the closest to the sink. First, node  $i$  sends its data  $Y_i$  to its neighbor  $i+1$ . In that node, the first step of DPCM is performed, namely, the signal  $Y_{i+1}$  is predicted using  $Y_i$ , and the corresponding (quantized) residual  $\hat{\epsilon}_{i+1}$  is calculated. Then, node  $i+1$  sends both the signal received from its neighbor and its own residual to node  $i+2$ . Node  $i+2$  first reconstructs signals  $Y_i$  and  $Y_{i+1}$ , and then calculates the next residual  $\hat{\epsilon}_{i+2}$  and so on. At the end, the leader-node of each group has the coded signal  $Y_i$ , and the



**Fig. 5.** Network segmentation with *AdaptNetHaar* for various positions of a single discontinuity (the sink is at the extreme right).



**Fig. 6.** Relative cost for *NetDPCM* as a function of network size ( $M$ ). Correlation:  $\beta = 2$  and  $\alpha \in \{0.0005, 0.001, 0.01\}$ .



**Fig. 7.** Network divisions for *NetDPCM* with  $\gamma = 10^{-4}$ . Network sizes:  $N = 512, 1024$  and  $4096$  sensors

set of residuals (innovations) corresponding to the rest of signals in the group. We call *NetDPCM* our algorithm. In terms of computation, we can see that for one step of prediction we need to perform two matrix multiplications and one matrix inversion, and for large groups, this implies a substantial computational complexity which will increase the power dissipated. Thus, the size of the groups will be limited due to the increasing value for  $\text{cost}_{DSP}$  relative to the total data gathering cost. By considering the number of additions and multiplications required for each of these operations, the processing cost can be written as follows [5]:

$$\text{num}_{OP}(N) = 2 \cdot \left( \sum_{j=2}^g j^2 + \sum_{j=2}^g j^3 \right) \quad (7)$$

where  $g$  is the number of sensors in the given group. Our simulations show that the optimal network division has group sizes that are larger as the distance to the sink increases, and successfully smaller as we approach the sink.

## 5. EXPERIMENTAL RESULTS

For *AdaptNetHaar*, we use piecewise constant signals with amplitude values uniformly distributed in the interval  $[-1, 1]$ . The number of signal discontinuities varies with the network size, and it is given<sup>3</sup> as  $\left\lceil \rho \frac{N}{\log_2 N} \right\rceil$ , with  $\rho$  a small real number  $\rho \in [0.1, 5]$ . For each number of discontinuities, the performance was obtained by averaging over 1000 realizations. For *NetDPCM*, we use the correlation structure with parameters  $\beta = 2$ ,  $\alpha \in \{0.0005, 0.001, 0.01\}$  and  $\gamma = 10^{-4}$ . The distance between each pair of neighboring sensors is fixed ( $d = 10$  units). Fig. 4 illustrates the cost gains of *AdaptNetHaar* for different values of the parameter  $\rho$ , and Fig. 5 presents some examples of optimal network division for four different particular signals. Fig. 6 illustrates the dependency of the gain on the network size for *NetDPCM*. We can see that for small enough values of the correlation parameter  $\alpha$  (i.e. strong enough correlation), we achieve a considerable power gain. Also, the algorithm gain increases with the network size. In Fig. 7, we illustrate the resulting group sizes for the algorithm *NetDPCM*.

## 6. REFERENCES

- [1] R. Cristescu, B. Beferull-Lozano, M. Vetterli, *On network correlated data gathering*, in Proc. INFOCOM'04.
- [2] A. Ciancio, A. Ortega, *A distributed wavelet compression algorithm for wireless sensor networks using lifting*, in Proc. ICASSP'04.
- [3] R. Gray, *Quantization*, IEEE Trans. on Information Theory, 1998.
- [4] G. J. Pottie, W. J. Kaiser, *Wireless integrated sensor networks*, Communications of ACM 2000.
- [5] A. Wang, A. Chandrakasan, *Energy Efficient DPSs for Wireless Sensor Networks*, IEEE Signal Processing Magazine, 2002.
- [6] N. A.C. Cressie, *Statistics for spatial data*, Wiley Series in Probability and Mathematical Statistics, 1991.

<sup>3</sup>The notation  $\lceil x \rceil$  denotes the integer part of  $x$ .